Event Detection in Aerospace Systems using Centralized Sensor Networks:
A Comparative Study of Several Methodologies

Ali Farhang Mehr [1, 2]
Research Scientist

Julien Sauvageon [3]
Graduate Student

Alice M. Agogino [3]
Professor

Irem Y. Tumer [1]
Group Lead

[1] Complex Systems Design and Engineering Group
Intelligent Systems Division
NASA Ames Research Center

[2] Corresponding Author: Voice: (650) 604-1140; E-mail: amehr@email.arc.nasa.gov

[3] Department of Mechanical Engineering
University of California, Berkeley

Abstract – Recent advances in micro electro-mechanical systems technology, digital electronics, and wireless communications have enabled development of low-cost, low-power, multifunctional miniature smart sensors. These sensors can be deployed throughout a region in an aerospace vehicle to build a network for measurement, detection and surveillance applications. Event detection using such centralized sensor networks is often regarded as one of the most promising health management technologies in aerospace applications where timely detection of local anomalies has a great impact on the safety of the mission. In this paper, we propose to conduct a qualitative comparison of several local event detection algorithms for centralized redundant sensor networks. The algorithms are compared with respect to their ability to locate and evaluate an event in the presence of noise and sensor failures for various node geometries and densities.

Keywords: Centralized Sensor Networks, Event Detection, IVHM, Data Fusion

I. INTROCUTION

The primary purpose of Integrated Vehicle Health Management (IVHM) is to increase safety and reliability of a mission-critical engineering system (e.g., an aerospace vehicle) while simultaneously reducing its maintenance costs. This is often accomplished via an onboard event detection engine that identifies system degradation or failures and takes appropriate mitigating steps accordingly [1, 2, 3, 4]. Event detection is one of the most promising applications of sensor networks where a large number of networked nodes are used to identify regions experiencing some particular phenomenon. While researchers have developed a variety of event detection techniques, the relative performance and robustness of such techniques seems to be largely unknown.

Therefore, in this paper, we aim to devise an objective framework for comparison of different centralized network event detection algorithms. We will investigate a variety of algorithms – ranging from computationally-complex techniques such as Mote-FVF (Fuzzy Validation and Fusion) algorithm [6,7] to simpler methods such as Model Fitting Interpolation, Polynomial Regression, and the Distributed Gaussian Method – and benchmark their ability to detect and locate local events, from the mass of sensors readings in an efficient and robust manner. These algorithms use the correlation between the sensors readings to ensure robustness (no model of the system is used) and can be applied to all types of dense networks.

II. APPROACH

The benchmark sensor network used in this paper is sensing the surface temperature of an aluminum plate. The local event is defined as a local rise of the temperature and is referred to as a hot spot. The algorithms' performances depend on the hot-spot location, the nodes repartition and their density. In order to take into account these factors, we consider four different network configurations (quadrants, triangles, random, semi-random) as illustrated Fig. 1,

with a number of nodes of 16, 36, 64 and 100. In the semi-random configuration, the region is divided into small parts with the same number of nodes in each part to ensure a better coverage.

Quadrants Triangles Random Semi-Random

Fig. 1. Network Geometries
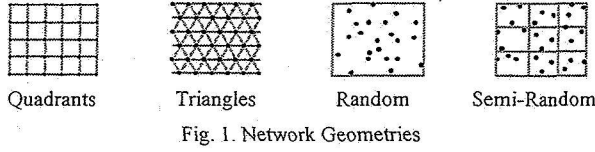
10 cases are generated with different hot-spot locations. Fig. 2 represents the case of a network with 6x6 nodes placed in quadrant.
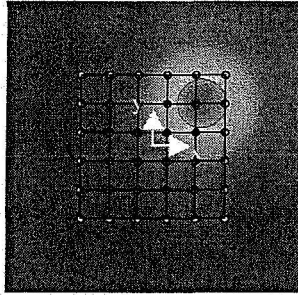

Fig. 2. Sensors Placement and Temperature Repartition in the Plate

A. What to Compare?

This paper studies the ability of the algorithms to locate precisely the temperature peak and to give a fair estimation of the peak value in the cases of noisy nodes and nodes failures. In order to compare the robustness of the algorithm, two cases are considered:

• In the first one, a ±2.5% and ±5% normally distributed noise is introduced within the readings. The mean value of the peak location, the standard deviation of the location, the mean peak value and its standard deviation are compared.

• In the second case, one or two failing sensor are introduced in the network (with a node value 10% and 15% higher or lower from the normal value) and the peak location and value compared with the ideal case.

B. How to Compare?

In order to evaluate all the different cases and synthesize all the results in a consistent way, we use a qualitative analysis. To assist in this approach, a display has been developed to visualize the principal characteristics of algorithms. The display, shown in Fig 3 at the end of the paper, is executed for each different case/algorithm to show the shape of the temperature field estimated by the algorithm. We will then measure the robustness of these algorithms by varying the peak locations and the peak values under noise and under failures. In the end, the studied algorithms are graded for their performance/robustness in detection the peak location and the peak values and absorbing variations in the peak location and faulty sensor readings.

III. ALGORITHMS

A. Model Fitting Interpolation

The method used here for the interpolation is the bicubic technique. It is one of the most common interpolation methods in two dimensions. With this method, the value of the function f(x, y) at a point (x, y) is computed as a weighted average of the nearest sixteen nodes. It is composed of two basic cubic interpolations put together, one for each plane direction. The interpolation is calculated with the Eq. 1.

$$T_{Int}(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} \alpha_{ij} x^i y^j \qquad [1]$$

The coefficients α_{ij} are computed by Matlab and depend on the interpolated data source properties.

B. Polynomial Regression

The polynomial used for the hot spot regression is a 4rth with the shape shown by Eq. 2:

$$T_{Reg}(x,y) = \sum_{i=0}^{4}\sum_{j=i}^{4} \beta_{ij} x^i y^{j-i} \qquad [2]$$

The coefficients β_{ij} are computed by Matlab using the least square method in order to minimize the Eq. 3 at the nodes.

$$S = \sum_{i=1}^{n} \left(T(x_i,y_i) - \hat{T}(x_i,y_i) \right) \qquad [3]$$

Where $\hat{T}(x_i,y_i)$ represents the real value of the temperature at the ith node and $T(x_i,y_i)$ the value given by the interpolation.

C. Distributed Gaussian Method

The idea of this method is to generate a Gaussian curve centered on each node and to do a normalized summation of all of them and then find the maximum to detect the Temperature peak. The Gaussian curve centered on the node is given by Eq. 4.

$$G_i(x,y) = \hat{T}(x_i, y_i) \times e^{-\frac{(x_i-x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i-y)^2}{2\varepsilon^2}} \qquad [4]$$

where ε is the standard deviation of the distribution and can be tuned depending on the application.

In order to reduce the geometric effect of the node placement, the summation of the Gaussian function have to be normalized as shown in Eq. 5.

$$T_{Gau}(x,y) = \frac{\sum_{i=1}^{n} \hat{T}(x_i, y_i) \times e^{-\frac{(x_i-x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i-y)^2}{2\varepsilon^2}}}{\sum_{i=1}^{n} e^{-\frac{(x_i-x)^2}{2\varepsilon^2}} \times e^{-\frac{(y_i-y)^2}{2\varepsilon^2}}} \qquad [5]$$

D. Mote Fuzzy Validation and Fusion Method (mote_fvf)

The Mote Fuzzy Validation and Fusion algorithm [6] was developed for wireless sensors network. It is able to distinguish between sensor failure and from environment abnormal behavior and to extract the relevant information from the mass of data of the sensor network. Methods for sensor validation and fusion based on fuzzy logic are unique as they do not require a mathematical model of the system. This algorithm uses the redundancy of the network to compensate the lack of reliability.

The network takes some redundant sensor readings and makes them go through three major steps: validation, fusion and prediction to come up with one single robust value. So far, this algorithm has been applied only to uniform fields where the fusion was done with all the sensors. In order to apply it to a non-uniform field (hot spot) the fusion is done locally between few sensors located in restricted areas where the field can be assumed uniform [7]. For this application, one local fused value is generated by 3 or 4 sensor values as illustrate Fig. 4.
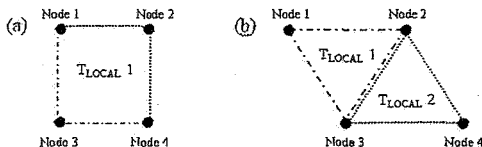

Fig. 4. Fused Local Values for (a) Quadrant Repartition (b) Triangle Repartition

The validation part of the algorithm first filters obvious failures based on sensors physical limitation. Then it finds the medium of all the readings by a majority voting system and finally generates a dynamic validation curve in order to assign a confidence value $\sigma_i \in [0 \quad 1]$ to the readings x_i. The center of the validation curve, where $\sigma = 1$, is a balance between the medium of the values and a prediction part. The fusion Eq. 6 consists of a weighted average of the values and their confidence values with include a fraction of the predicted value $\hat{x}$ to prevent the system from becoming unstable:

$$x_f = \frac{\sum_{1}^{n} x_i \sigma(x_i) + \frac{\alpha \hat{x}}{\omega}}{\sum_{1}^{n} \sigma(x_i) + \frac{\alpha}{\omega}} \qquad [6]$$

The prediction part is an exponential weighted moving average time series predicting method. Finally, the robust fused values obtained at the end of the process are interpolated to have the shape of the temperature field.

IV. COMPARISON METHODOLOGY

In order to explain the method used to compare different algorithms for different cases, this section focuses on the study of one particular network configuration, quadrant repartition with 6x6 nodes. By processing the algorithms for 10 different hot spot location, it appears that the behaviors of the algorithms are a combination of 3 basic ones represented in Fig. 5.

(S1) middle of a quadrant, (S2) middle of two nodes, (S3) centered in one node
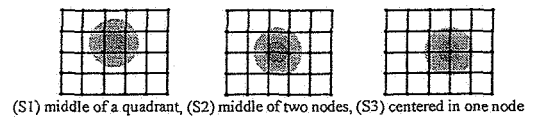
Fig. 5. Basic Hot Spot Location

By devising several metrics to measure performance/robustness in various cases, we have graded the algorithms from 5 for good to 0 for poor. Afterwards, the values of the different cases are summed to provide a global comparison. A summary of the results is presented Table 1 & 2 at the end of the paper (Appendix I).

IV. RESULTS

The same comparison methodology is applied to several different network designs.

A. Noise

The results of the comparison in the presence of noise are presented in the form of a matrix of charts, Fig. 6 (Appendix I). Each line represents a different configuration and each column a different algorithm. Each plot of the matrix represents the performance parameters with the number of nodes in the network. The blue line is peak value performance, the red one is the standard deviation of the peak location and the black line represents the location performances. The random repartition is not presented here as its performances are really low.

, The variation of the peak value due to the noise is not represented in the matrix. It appears that this parameter is fairly independent of the geometry and the density of the network. The effect of a 5% noise on the variation of the peak value is presented Table 3.

Table 3. Variation of the Peak Value due to a 5% Noise

	Interpolation	Regression	Gaussian	Mote_fvf
5% noise	4.1%	0.7%	1.3%	1.5%

B. Sensor Failure

The effect of the failures depends a lot of their location. The results presented in the Table 4 are divided into two parts whether if the failure is inside or out side of the hot spot. The values presented in the table are not the absolute performance of the algorithm but are relative to the ideal case without failure. The geometry of the network and the density do not influence these values.

Table 4. Evaluation of the Algorithm with one Faulty Sensor

		Interpolation	Regression	Gaussian	Mote_fvf
Failure Outside	Peak Location	0	12	12	15
	Peak Value	0	11	11	15
Failure Inside	Peak Location	0	7	8	13
	Peak Value	0	9	9	13

V. OBSERVATIONS

As expected, increasing the number of nodes in the network increases the performances of the algorithms. Also, as it was supposed, the geometry of the network changes the performances. A good repartition of the nodes through the all area gives much better results than a random or a semi-random placement. The difference between the quadrant and triangle reparation is just sensible for the mote_fvf.

In the case of a noisy environment without sensor failure, a simple interpolation gives fairly good results while the regression and Gaussian algorithm have some difficulties to estimate the value of the peak.

With respect to fault tolerance of various algorithms, the mote_fvf algorithm is the most robust. This algorithm can handle a large number of failures. Moreover, the Mote_fvf performs robustly regardless of where the faulty sensor is located whereas other algorithms' performances depend on the location of the fault.

VI. RECOMMANDATIONS

If the probability of a faulty sensor is negligible, one may choose to use Model Fitting Interpolation (low computational complexity). The shape of the interpolation function can be modified to better fit the application. But this method is very sensitive to sensor faults. The presence of a faulty sensor will cause a wrong diagnosis.

In the presence of faults, Mote Fuzzy Validation and Fusion algorithm is the most robust algorithm to use. The price of this robustness is an increase in the computation time by a factor of 8 to 16. The mote_fvf works the best in a triangular geometry. Nevertheless if the probability of failure is very high, quadrant repartition can be used to increase the robustness as the algorithm by fusing the value of 4 nodes.

REFERENCES

[1] K. Datta, N. Jize, D. Maclise, D. Goggin. An IVHM Systems Analysis and Optimization Process. 2003, IEEEAC Paper No. 1094.

[2] D. Abbott, et.al. 2002. "Development and Evaluation of Sensor Concepts for Ageless Aerospace Vehicles: Development of Concepts for an Intelligent

Sensing System", NASA Report no. NASA/CR-2002-211773.

[3] M. Hedley, M. Johnson, C. Lewis, D. Carpenter, H. Lovatt, D. Price. "Smart Sensor Network for Space Vehicle Monitoring". In Proceedings of the International Signal Processing Conference, Dallas TX, March 2003.

[4] D.C. Price et.al. "An Integrated Health Monitoring System for an Ageless Aerospace Vehicle". In Proceedings of the International Workshop on Structural Health Monitoring, Stanford, September 2003.

[5] T.J. Johnson et al., "Distributed structural health monitoring with a smart sensor array", Mechanical Systems and Signal Processing 18 (2004) pp.555–572

[6] Y.-J Wen, A.M. Agogino and K. Goebel, "Fuzzy Validation and Fusion for Wireless Sensor Networks", Proceedings of the ASME International Mechanical Engineering Congress,Anaheim, CA, 2004.

[7] J. Sauvageon, "Integrated Systems Health Monitoring Using Smart Dust Mote Sensor Networks: Hot Spot and Peak Strain Detection in Space Vehicles", Master Report 2005, UC Berkeley
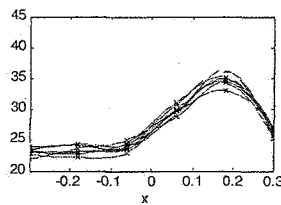
Appendix I

Fig. 3. Plots Format used to Compare the Algorithms

Table 1. Comparison of the Algorithms Hot Spot Location Detections

		(S1)				(S2)				(S3)			
		Peak Location		Peak Value		Peak Location		Peak Value		Peak Location		Peak Value	
		Center	Std V	Value	Std V	Center	Std V	Value	Std V	Center	Std V	Value	Std V
Interpolation	noise	5	2	5	1	5	2	5	1	5	5	5	1
	fault	1		0		1		0		4		0	
Regression	noise	5	4	2	4	5	4	1	4	5	4	0	4
	fault	3		2		2		1		3		2	
Gaussian	noise	5	3	2	3	5	4	1	3	5	5	0	3

	fault	3		2		2		1		3		2	
Mote_fvf	noise	5	3	4	3	5	4	5	3	5	3	2	3
	fault	5		4		5		4		5		4	

Table 2. Results of the Algorithms Comparison

		Results			
		Peak Location		Peak Value	
		Center	Std V	Value	Std V
Interpolation	noise	15	9	15	3
	fault	6		0	
Regression	noise	15	12	3	12
	fault	8		5	
Gaussian	noise	15	12	3	9
	fault	8		5	
mote_fvf	noise	15	10	11	9
	fault	15		12	

Interpolation   Regression   Gaussian   Mote_fvf

Quadrant

Triangle

Semi-Random

— Peak Value Performance
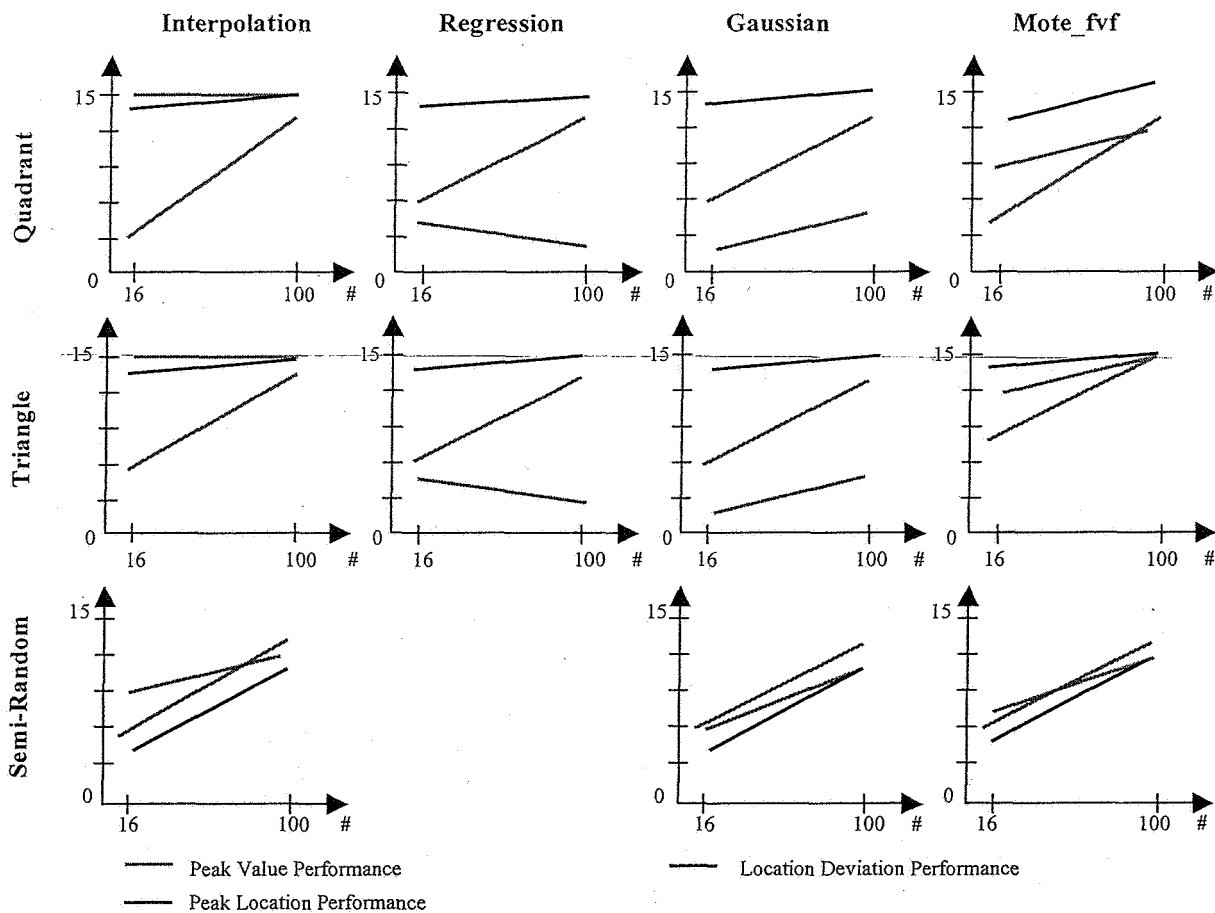— Peak Location Performance
— Location Deviation Performance

Fig. 6: Comparison of Hot Spot Detection Algorithm Under Noise